

11/09/99
jc604 U.S. PTO

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship..... Howard et al.
Applicant..... Microsoft Corporation
Attorney's Docket No. MS1-379US
Title: Methods and Systems for Screening Input Strings Intended for Use by Web Servers

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks
Washington, D.C. 20231

From: Lance R. Sadler (509) 324-9256
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

1. Transmittal Letter with Certificate of Mailing included.
2. PTO Return Postcard Receipt
3. New patent application (title page plus 23 pages, including claims 1-31 & Abstract)
4. Executed Declaration
5. 4 sheets of formal drawings (Figs. 1-4)
6. Assignment w/Recordation Cover Sheet

Large Entity Status [x]

Small Entity Status []

The Commissioner is hereby authorized to charge payment of fees or credit overpayments to Deposit Account No. 50-0463 in connection with any patent application filing fees under 37 CFR 1.16, and any processing fees under 37 CFR 1.17.

Date:

11/9/99

By:

Lance R. Sadler
Lance R. Sadler
Reg. No. 38,605

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable)

EL427262845

Date:

11/9/99

By:

Dana L. Calhoun
Dana L. Calhoun

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Methods and Systems for Screening Input Strings
Intended for Use by Web Servers**

Inventor(s):
Michael Howard
Vikas Malhotra

ATTORNEY'S DOCKET NO. MS1-379US

1 **TECHNICAL FIELD**

2 This invention relates to methods and systems for screening input strings
3 that are intended for use by Web servers. In particular, the invention pertains to
4 methods and systems for identifying input strings that contain attack patterns that
5 can be used to attack a Web server, and, in some instances, reacting to the attack
6 patterns once identified.

7
8 **BACKGROUND**

9 Web servers are computers that are used to provide access to various
10 resources, e.g. Web pages, for various client devices such as browsers. Typically,
11 an individual uses a client device to provide an input string, such as a URL, to the
12 Web server. The URL indicates to the Web server the location of the particular
13 resource of interest. The Web server then locates the resource using the URL and
14 returns the resource to the client device so that it can be displayed for the
15 individual. Other types of input strings can be provided to the Web server by the
16 client, e.g. input strings in the form of HTTP verb requests (e.g. POST requests)
17 including WebDAV requests.

18 In the past, malicious individuals have used input strings that are intended
19 for use by Web servers to attack the servers. These individuals will typically try to
20 find an input string that causes the Web server or, perhaps its operating system, to
21 perform in a manner that is inconsistent with simply processing legitimate client
22 requests and returning authorized resources to the client. Input strings that have
23 been used in the past to attack Web servers seem to come in an ever-changing
24 number of varieties and formats. The various attacks that can be waged against a
25

1 Web server can be categorized as disclosure attacks, integrity attacks, and denial
2 of service attacks.

3 A disclosure attack takes place when an individual attacks a web site and
4 attempts to read information that they are not authorized to read. For example,
5 there may be some executable code at the server that an individual is not
6 authorized to view. Yet, by providing an input string that causes the server to
7 malfunction, the individual actually gets to view the executable code. Consider,
8 for example, Active Server Pages. Active Server Pages can allow Web developers
9 to use scripting languages like Visual Basic Script and JScript to pass information
10 to various components that contain logic for accessing databases, instruct the
11 components to perform a programmed action, and return the results of the
12 programmed action. The individual is only authorized, and supposed to view the
13 results of the programmed action. Yet, by using particular inappropriate input
14 strings it may be possible for the individual to view the code that produces the
15 results.

16 An integrity attack is similar to a disclosure attack in that an individual can
17 gain access to unauthorized information. In addition to gaining access to the
18 information, however, integrity attacks involve the manipulation of data or
19 information that is being viewed. This is particularly problematic because the
20 changed, now-invalid information can potentially further compromise an already-
21 compromised Web server.

22 A denial of service attack is an attack that can cause a decrease in the
23 quality of service or, ultimately, can cause the server to crash. This can adversely
24 impact the server's ability to service other legitimate clients thereby leading to
25 undesirable downtime and customer dissatisfaction.

1 Many of these types of attacks can be traced directly to the mishandling of
2 an input string that was provided to the Web server. A need exists to deal with
3 problematic input strings in a flexible, quick and convenient manner.
4 Accordingly, this invention arose out of concerns associated with providing
5 improved methods and systems for recognizing problematic input strings and
6 dealing with them before they adversely affect the Web server.

7 8 **SUMMARY**

9 Methods and systems of screening input strings that are intended for use by
10 a Web server are described. In the described embodiment, an attack pattern is
11 determined that can be used to attack a Web server. A search pattern is defined
12 that can be used to detect the attack pattern. The search pattern is defined in a
13 flexible, extensible manner that permits variability among its constituent parts. An
14 input string that is intended for use by a Web server is received and evaluated
15 using the search pattern to ascertain whether the attack pattern is present. If an
16 attack pattern is found that matches the search pattern, then a remedial action is
17 implemented.

18 19 **BRIEF DESCRIPTION OF THE DRAWINGS**

20 Fig. 1 is a block diagram of a client/server network system having a client
21 and server.

22 Fig. 2 is a block diagram of an exemplary computer that can be used to
23 implement the client and/or the server of Fig. 1.

24 Fig. 3 is a flow diagram that describes steps in a method in accordance with
25 an embodiment of the invention.

1 Fig. 4 is a block diagram of an input string screening tool in accordance
2 with an embodiment of the invention.

3 4 **DETAILED DESCRIPTION**

5 **Network Configuration**

6 As a preliminary matter, the following disclosure assumes a familiarity with
7 Internet and WWW practices, formats, and protocols. A great number of books
8 are available on these subjects. Stout, Rick, *The World Wide Web: Complete*
9 *Reference*, McGraw-Hill, 1996, is one example.

10 Fig. 1 shows an information server system 12 connected for data
11 communication with associated clients or client devices 14. The information
12 server system comprises a server 16 and a repository 18 of published data and
13 other content. Server 16 is a computer that executes one or more server programs
14 17. Server 16 has access to repository 18, which is typically a hard disk or other
15 mass storage device. Mass storage device 18 can be located either locally or
16 remotely, and can be connected through a local bus, a local-area network, or a
17 wide-area network such as the Internet. Server 16 includes a data processor,
18 electronic memory, and other components common to computers that are used for
19 server applications.

20 Each of client devices 14 is any device such as a personal computer that
21 might be connected to receive information from server system 12. Client device
22 14 has a client viewer or browser 20 that forms an interface to a human user or
23 operator. Client viewer 20 interprets instruction and data streams provided by
24 system 12 and in response presents information to the user in textual, graphical, or
25 other forms. Client viewer 20 also accepts operator input, and allows the operator

1 to select and navigate from one hypermedia document to another using hyperlinks
2 as described above. Client viewer 20 is connected for data communications with
3 server system 12 by a suitable communications medium such as a local or wide
4 area network, by a modem link, or by a similar continuous or on-demand
5 connection. Data connection can be made between server system 12 and client
6 devices 14 through the Internet, using a standard protocol, such as HTTP
7 (hypertext transport protocol).

8 An exemplary client viewer 20 is a conventional, off-the-shelf Internet Web
9 browser, having features and functions such as are common to popular Web
10 browsers. Client viewer 20 is not limited to any particular type of Web browser.
11 For instance, client viewer 20 might be the Internet Explorer, available from
12 Microsoft Corporation of Redmond, Washington, or a Netscape Navigator
13 browser, available from Netscape of Mountain View, California. Each of these
14 browsers supports a different feature set, and responds to different commands and
15 command sets. In addition, the term "client viewer" as used herein encompasses
16 any software that is used by a client to interpret data obtained from server system
17 12. In the future, such software will likely comprise a variety of downloadable
18 components and helper applications used by software other than traditional
19 browsers to render multimedia content from the Internet or other servers.

20 When a user wishes to access a resource that is accessible through the
21 information server system 12, a data stream or input string, e.g. a URL, is prepared
22 by their client viewer 20 and sent to the information server system 12 via an
23 appropriate connective network. The information server system 12 receives the
24 request, processes it, and returns the requested resource to the client 14. The client
25 viewer 20 then enables the user to view the requested resources. Other input

1 strings can be prepared and sent from the client viewer 14 to the information
2 server system 12. An exemplary input string is one that is associated with an
3 HTTP verb request, such as a POST request. Of course, other input strings can be
4 utilized.

6 **Exemplary Computer Architecture**

7 Fig. 2 shows a general example of a computer 130 that can be used to
8 implement the client 14 and/or the server 12.

9 Computer 130 includes one or more processors or processing units 132, a
10 system memory 134, and a bus 136 that couples various system components
11 including the system memory 134 to processors 132. The bus 136 represents one
12 or more of any of several types of bus structures, including a memory bus or
13 memory controller, a peripheral bus, an accelerated graphics port, and a processor
14 or local bus using any of a variety of bus architectures. The system memory 134
15 includes read only memory (ROM) 138 and random access memory (RAM) 140.
16 A basic input/output system (BIOS) 142, containing the basic routines that help to
17 transfer information between elements within computer 130, such as during start-
18 up, is stored in ROM 138.

19 Computer 130 further includes a hard disk drive 144 for reading from and
20 writing to a hard disk (not shown), a magnetic disk drive 146 for reading from and
21 writing to a removable magnetic disk 148, and an optical disk drive 150 for
22 reading from or writing to a removable optical disk 152 such as a CD ROM or
23 other optical media. The hard disk drive 144, magnetic disk drive 146, and optical
24 disk drive 150 are connected to the bus 136 by an SCSI interface 154 or some
25 other appropriate interface. The drives and their associated computer-readable

1 media provide nonvolatile storage of computer-readable instructions, data
2 structures, program modules and other data for computer 130. Although the
3 exemplary environment described herein employs a hard disk, a removable
4 magnetic disk 148 and a removable optical disk 152, it should be appreciated by
5 those skilled in the art that other types of computer-readable media which can
6 store data that is accessible by a computer, such as magnetic cassettes, flash
7 memory cards, digital video disks, random access memories (RAMs), read only
8 memories (ROMs), and the like, may also be used in the exemplary operating
9 environment.

10 A number of program modules may be stored on the hard disk 144,
11 magnetic disk 148, optical disk 152, ROM 138, or RAM 140, including an
12 operating system 158 (e.g., the server operating system 22 below), one or more
13 application programs 160 (e.g., application(s) 30, Internet Information Server 24
14 below), other program modules 162 (e.g., platform 26 below), and program data
15 164. A user may enter commands and information into computer 130 through
16 input devices such as a keyboard 166 and a pointing device 168. Other input
17 devices (not shown) may include a microphone, joystick, game pad, satellite dish,
18 scanner, or the like. These and other input devices are connected to the processing
19 unit 132 through an interface 170 that is coupled to the bus 136. A monitor 172 or
20 other type of display device is also connected to the bus 136 via an interface, such
21 as a video adapter 174. In addition to the monitor, personal computers typically
22 include other peripheral output devices (not shown) such as speakers and printers.

23 Computer 130 commonly operates in a networked environment using
24 logical connections to one or more remote computers, such as a remote computer
25 176. The remote computer 176 may be another personal computer, a server, a

1 router, a network PC, a peer device or other common network node, and typically
2 includes many or all of the elements described above relative to computer 130,
3 although only a memory storage device 178 has been illustrated in Fig. 2. The
4 logical connections depicted in Fig. 2 include a local area network (LAN) 180 and
5 a wide area network (WAN) 182. Such networking environments are
6 commonplace in offices, enterprise-wide computer networks, intranets, and the
7 Internet.

8 When used in a LAN networking environment, computer 130 is connected
9 to the local network 180 through a network interface or adapter 184. When used
10 in a WAN networking environment, computer 130 typically includes a modem 186
11 or other means for establishing communications over the wide area network 182,
12 such as the Internet. The modem 186, which may be internal or external, is
13 connected to the bus 136 via a serial port interface 156. In a networked
14 environment, program modules depicted relative to the personal computer 130, or
15 portions thereof, may be stored in the remote memory storage device. It will be
16 appreciated that the network connections shown are exemplary and other means of
17 establishing a communications link between the computers may be used.

18 Generally, the data processors of computer 130 are programmed by means
19 of instructions stored at different times in the various computer-readable storage
20 media of the computer. Programs and operating systems are typically distributed,
21 for example, on floppy disks or CD-ROMs. From there, they are installed or
22 loaded into the secondary memory of a computer. At execution, they are loaded at
23 least partially into the computer's primary electronic memory. The invention
24 described herein includes these and other various types of computer-readable
25 storage media when such media contain instructions or programs for implementing

1 the steps described below in conjunction with a microprocessor or other data
2 processor. The invention also includes the computer itself when programmed
3 according to the methods and techniques described below.

4 For purposes of illustration, programs and other executable program
5 components such as the operating system are illustrated herein as discrete blocks,
6 although it is recognized that such programs and components reside at various
7 times in different storage components of the computer, and are executed by the
8 data processor(s) of the computer.

10 **Input String Screening**

11 Aspects of the invention enable an input string that is provided by a client
12 to be screened before it is processed by the Web server. An “input string” is a
13 URL or other string that is intended for use by the Web server. Screening the input
14 strings ensures that problematic input strings are identified and handled
15 appropriately so that the risk of adversely impacting the Web server is reduced.
16 As an example of a problematic input string consider the following URL input
17 string:

18
19 <http://www.foo.com/../../../boot.ini>
20

21 Assume that data that is associated with www.foo.com is stored in a
22 directory “c:\wwroot\stuff\data”. The “..” that appears in the URL input string
23 after the www.foo.com specification can cause the server to move up in the
24 hierarchical directory from “c:\wwroot\stuff\data” by one directory. A series of
25 “..” in the URL input string can cause the server to move up in the hierarchical

1 directory a number of times until it reaches the root directory, in this case the “c:”
2 directory. At this point it might be possible to get access any files in the root
3 directory such as the specified “boot.ini” file. This file might constitute a file that
4 describes how the computer is designed to boot. In this case, a user would be able
5 to view and possibly manipulate an unauthorized file. As another example,
6 consider the following URL input string:

7
8 [http://www.foo.com/datalookup.asp::\\$DATA](http://www.foo.com/datalookup.asp::$DATA)
9

10 In this example, it is possible that the server might not understand the
11 “::\$DATA” portion of this input string, but that the string portion has a special
12 meaning to the operating system on which the server is executing. As a
13 consequence, the operating system might cause unauthorized files to be accessible
14 to the user.

15 In both of these examples, the input string can be characterized as
16 containing a pattern that is problematic to the Web server. It is problematic
17 because it can cause the Web server or its operating system to behave in a manner
18 that is inconsistent with returning only authorized resources to a client. In this
19 document, such patterns are referred to as “attack patterns” because they
20 effectively enable an attack on the server. In the above two examples, the attack
21 patterns are constituted by the “..” and “::\$” portions of the input string.

22 In addition to these exemplary attack patterns, there are also input string
23 characteristics that can be indicative of an attack pattern. One such characteristic
24 is if the input string does not contain an alphabetical character at its end. Another
25 characteristic is whether the input string contains any specific “operators” that are

1 inappropriate for an input string. Examples include the operators “|”, “<”, “>”,
2 and “&”. Any input string that is found to satisfy the characteristics that are
3 indicative of an attack pattern are likely to be problematic for the server.
4

5 **Web Server Pattern Matching**

6 Fig. 3 shows a flow diagram that describes steps in an input string
7 screening method for a Web server in accordance with one embodiment of the
8 invention. Step 200 determines an attack pattern that can be used to attack a Web
9 server. One way in which this determination can be made is by simply observing
10 over time, which attacks on a Web server are successful. Another way to
11 determine an attack pattern is to recognize that there are input string characteristics
12 that can be problematic for a Web server. For example, input strings that contain
13 the pattern “..” can be problematic because they might enable an individual to
14 inappropriately “walk” up a directory tree. Additionally, attack patterns can be
15 determined by recognizing that there are certain characters that are simply not
16 appropriate for inclusion in an input string. Examples of certain operators were
17 given above.

18 With one or more attack patterns having been determined, step 202 defines
19 a search pattern that can be used to detect the attack pattern. A search pattern is an
20 expression that is compared with input strings to determine whether there is a
21 matching search pattern in the input string. In the described embodiment, a search
22 pattern can be formatted syntactically in a manner that allows specification of both
23 identity and variability among constituent parts of an input string. Thus, the
24 search pattern can include literal parts that call for an exact character-by-character
25 match between those parts and corresponding parts of the input string, and

variable parts that allow for inexact matches or no match at all between those parts and corresponding parts of the input string. An input string is said to “match” a search pattern if the search pattern is found anywhere within the input string as specified by the search pattern. In the described embodiment, one or more search patterns are specified as regular expressions. In a regular expression, each character matches itself, unless it is one of a number of special characters that indicate variable characters in the input string. An example subset of regular expression definitions and their meanings is given below:

Pattern	Meaning
.	Matches an arbitrary character
(...)	Groups a series of pattern elements to a single element
^	Matches the beginning of the target
+	Matches the preceding pattern elements one or more times. For example, ba+c matches bac, baac, but not bc.
\$	Matches the end of the line. For example, 100\$ matches 100 at the end of a line.
[...]	Denotes a class of characters to match; [^...] negates the class. For example, b[aeiou]d matches bad, bed, bid, bod, and bud (but not bead or bead); and r[eo]+d matches red, rod, reed, rood, reod, roed, reood, roeod, etc.
[^]	Matches any character except those following the caret (^) character in the brackets, or any of an ASCII range of characters separated by a hyphen (-). For example, x[^0-9] matches xa, xb, xc, and so on, but not x0, x1, x2, and so on.
(... ...)	Matches one of the alternatives
?	Matches the preceding character zero or one time.
*	Matches the preceding character zero or more times. For example, ba*c matches bc, bac, baac, and so on.
{}	Matches any sequence of characters between the escaped braces. For example, {ju}+fruit matches jufruit, jujufruit, but not ufruit, jfruit, or ujfruit.
\	Removes the pattern match characteristics from the special characters listed above. For example, 100\$ matches 100 at the end of a line, but 100\\$ matches the character string 100\$ anywhere on a line.

By defining search patterns as described above, flexibility and extensibility are enhanced by enabling a system administrator to define a search pattern in terms of a generalized regular pattern that reflects an attack pattern of which the system administrator has recently become aware. The definition of search patterns

1 in this manner is timely because the search patterns can be defined almost as soon
2 as the attack patterns are detected, without the need to hardcode specific patterns.

3 In the described embodiment, patterns can be collected into collections of
4 patterns as more and more patterns are observed or determined. Accordingly, step
5 204 adds the pattern defined in step 202 to such a collection. The collection of
6 patterns can be stored and maintained in memory. In the described embodiment,
7 the collection is adapted for addition to, deletion of, or modification of the patterns
8 that it contains. This facilitates the overall extensibility of the collection of
9 patterns. In the described embodiment, steps 200-204 can be implemented using
10 an administrative tool or some other suitable interface.

11 Step 206 receives an input string from the client that is intended for use by
12 the Web server, and step 208 evaluates the input string using one or more of the
13 search patterns. Step 210 determines whether any of the attack patterns are
14 present in the input string. An attack pattern is present if a match is found for the
15 search pattern in the input string. If there are no attack patterns present in the
16 input string, then step 212 processes the input string or request that is associated
17 with the input string. Where an input string comprises a URL, processing can
18 include retrieving an appropriate resource, i.e. a Web page, and returning it to the
19 client. If, on the other hand, there is an attack pattern that is identified to be
20 associated with the input string (i.e. an attack pattern is found in the input string
21 that matches the search pattern), then step 214 implements a remedial action.
22 Remedial actions can be any actions that are associated with minimizing or
23 eliminating the effect that an attack pattern can have on the Web server. In but one
24 example, this can include denying a request that is associated with the input string.

1 For example, in the case of an input string that is a URL, this could mean returning
2 an error message to the client to the effect that the request could not be executed.

3 4 **Input String Screening Tool**

5 Fig. 4 shows an embodiment of an input string screening tool 300. The
6 illustrated input string screening tool 300 can be implemented in any suitable
7 software, hardware or firmware. In addition, the tool 300 can comprise an integral
8 part of a Web server, e.g. part of the Web server's parsing engine/function, or can
9 be implemented as an extension to an existing Web server. As an example, the
10 tool 300 can be implemented as an Internet Service Application Programming
11 Interface (ISAPI) extension that is suitable for use with Microsoft's Internet
12 Information Service (IIS) product. In this case, the ISAPI extension can register
13 with IIS to receive input strings when they are sent from a client. The ISAPI
14 extension then evaluates the input strings and determines whether any of the
15 defined attack patterns are present. If any attack patterns are determined to be
16 present, then the ISAPI extension can take any remedial action that is appropriate
17 in order to eliminate the risk to IIS.

18 In the Fig. 4 embodiment, input string screening tool 300 includes a pattern
19 matching engine 302 and a memory location 304. Memory location 304 contains
20 one or more patterns that have been defined and make up a pattern collection 306.
21 The patterns are stored in the memory location and are accessible to the screening
22 tool for evaluating input strings. The pattern matching engine can retrieve one or
23 more search patterns and use them to evaluate an input string to determine whether
24 it likely constitutes an attack on the Web server.

1 Advantages of the described embodiment include the ability to flexibly
2 define a plurality of extensible patterns that can be used to screen input strings for
3 attack patterns that can adversely affect performance of a Web server. System
4 administrators are given the opportunity to quickly take action by being able to
5 quickly define generalized regular patterns on the fly. This, in turn, increases the
6 response time so that any effects on a Web server that are associated with an attack
7 are mitigated to the extent possible.

8 Although the invention has been described in language specific to structural
9 features and/or methodological steps, it is to be understood that the invention
10 defined in the appended claims is not necessarily limited to the specific features or
11 steps described. Rather, the specific features and steps are disclosed as preferred
12 forms of implementing the claimed invention.
13
14
15
16
17
18
19
20
21
22
23
24
25

CLAIMS

1 **1.** A Web server input string screening method comprising:
2 determining an attack pattern that can be used to attack a Web server;
3 defining a search pattern that can be used to detect the attack pattern, the
4 search pattern being defined in a manner that permits variability among its
5 constituent parts;
6 receiving an input string that is intended for use by a Web server;
7 evaluating the input string using the search pattern to ascertain whether the
8 attack pattern is present; and
9 implementing a remedial action if an attack pattern is found that matches
10 the search pattern.

11
12
13 **2.** The Web server input string screening method of claim 1, wherein:
14 said defining comprises defining a plurality of different search patterns; and
15 said evaluating comprises evaluating the input string using said plurality of
16 different search patterns.

17
18 **3.** The Web server input string screening method of claim 1, wherein the
19 search pattern is specified as a regular expression.

20
21 **4.** The Web server input string screening method of claim 1, wherein
22 said receiving of the input string comprises receiving a URL.
23
24
25

1 **5.** The Web server input string screening method of claim 1, wherein
2 said receiving of the input string comprises receiving a portion of an HTTP verb
3 request.

4
5 **6.** The Web server input string screening method of claim 1, wherein
6 said implementing comprises denying a request that is associated with the input
7 string.

8
9 **7.** A Web server input string screening method comprising:
10 defining one or more search patterns that comprise literal characters and
11 special characters, wherein the literal characters indicate exact characters in an
12 input string that is intended for receipt by a Web server, and the special characters
13 indicate variable characters in an input string that is intended for receipt by the
14 Web server, the search patterns being usable to search for an attack pattern that
15 can be used to attack the Web server; and

16 storing the one or more search patterns in a memory location that is
17 accessible to a screening tool for evaluating an input string that is intended for
18 receipt by the Web server.

19
20 **8.** The Web server input string screening method of claim 7 further
21 comprising:

22 retrieving a search pattern from the memory location; and

23 evaluating an input string with the screening tool by ascertaining whether
24 the input string includes at least a portion that matches the search pattern.
25

1 **9.** The Web server input string screening method of claim 8, wherein the
2 evaluating of the input string comprises evaluating a URL.

3
4 **10.** The Web server input string screening method of claim 8, wherein
5 the evaluating of the input string comprises evaluating a portion of an HTTP verb
6 request.

7
8 **11.** The Web server input string screening method of claim 7 further
9 comprising implementing the screening tool as an extension for an existing Web
10 server.

11
12 **12.** The Web server input string screening method of claim 7 further
13 comprising implementing the screening tool as an ISAPI extension.

14
15 **13.** A Web server input string screening method comprising:
16 defining one or more search patterns that are specified as a regular
17 expression, the search patterns being usable to search for an attack pattern that can
18 be used to attack the Web server; and

19 storing the one or more search patterns in a memory location that is
20 accessible to a screening tool for evaluating an input string that is intended for
21 receipt by the Web server.

22
23 **14.** The Web server input string screening method of claim 13 further
24 comprising:

25 retrieving a search pattern from the memory location; and

1 evaluating an input string with the screening tool by ascertaining whether
2 the input string includes at least a portion that matches the search pattern.

3
4 **15.** The Web server input string screening method of claim 14, wherein
5 the evaluating of the input string comprises evaluating a URL.

6
7 **16.** The Web server input string screening method of claim 14, wherein
8 the evaluating of the input string comprises evaluating a portion of an HTTP verb
9 request.

10
11 **17.** A computer-readable medium having computer-readable
12 instructions thereon which, when executed by a computer, perform the method of
13 claim 14.

14
15 **18.** A Web server input string screening tool comprising:
16 a pattern matching engine that is configured to receive an input string that
17 is intended for use by a Web server and evaluate the input string to ascertain
18 whether it likely constitutes an attack on the Web server; and

19 one or more patterns that are usable by the pattern matching engine to
20 evaluate the input string, the patterns being defined in a manner that permits
21 variability among the constituent parts of the one or more patterns.

22
23 **19.** The Web server input string screening tool of claim 18, wherein the
24 one or more patterns are specified as regular expressions.

1 **20.** The Web server input string screening tool of claim 18, wherein the
2 pattern matching engine is configured to receive an input string that comprises a
3 URL.

4
5 **21.** The Web server input string screening tool of claim 18, wherein the
6 pattern matching engine is configured to receive an input string that comprises a
7 portion of an HTTP verb request.

8
9 **22.** One or more computer readable media having computer-readable
10 instructions thereon which, when executed by a computer perform the following
11 steps:

12 receiving an input string that is intended for use by a Web server;

13 evaluating the input string using a search pattern to ascertain whether the
14 input string contains an attack pattern that can be used to attack the Web server,
15 the search pattern comprising literal characters and special characters, wherein
16 literal characters indicate exact characters in the input string, and the special
17 characters indicate variable characters in the input string; and

18 implementing a remedial action if an attack pattern is found that matches
19 the search pattern.

20
21 **23.** The computer-readable media of claim 22, wherein said
22 implementing comprises denying a request that is associated with the input string.

1 **24.** The computer-readable media of claim 22, wherein said receiving
2 comprises receiving a URL.

3
4 **25.** The computer-readable media of claim 22, wherein said receiving
5 comprises receiving an input string that is associated with an HTTP verb request.

6
7 **26.** A collection of Web server screening patterns comprising:
8 a memory; and
9 a plurality of patterns stored in the memory, the patterns being useable to
10 screen input strings that are intended for use by a Web server, individual patterns
11 being defined in a manner that permits variability among their constituent parts.

12
13 **27.** The collection of claim 26, wherein the patterns are specified as
14 regular expressions.

15
16 **28.** The collection of claim 26, wherein the collection is adapted for
17 addition to, deletion of, or modification of patterns.

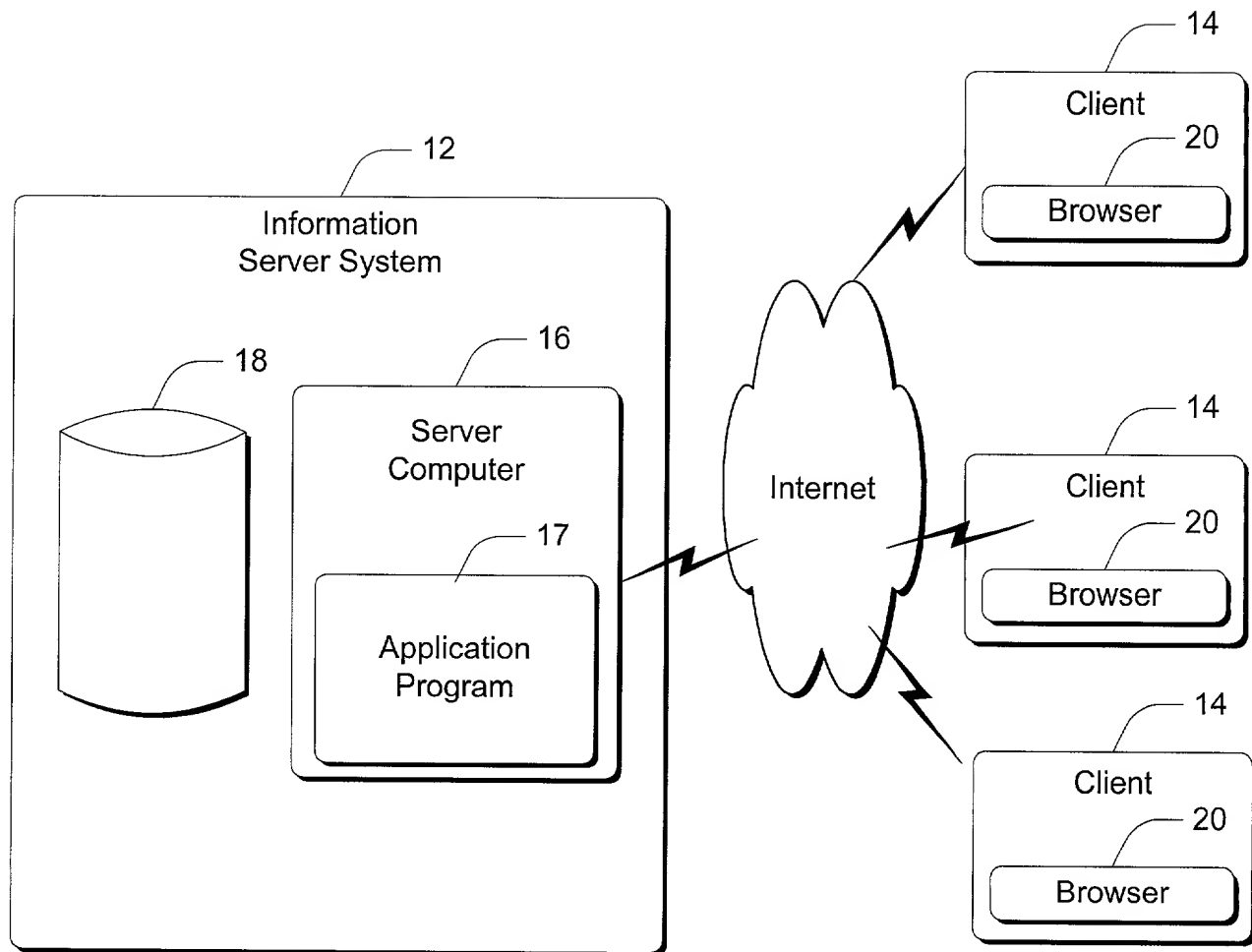
18
19 **29.** The collection of claim 26, wherein the patterns are configured for
20 use in screening URLs that are intended for use by a Web server.

21
22 **30.** The collection of claim 26, wherein the patterns are configured for
23 use in screening input strings associated with HTTP verb requests that are
24 intended for use by a Web server.
25

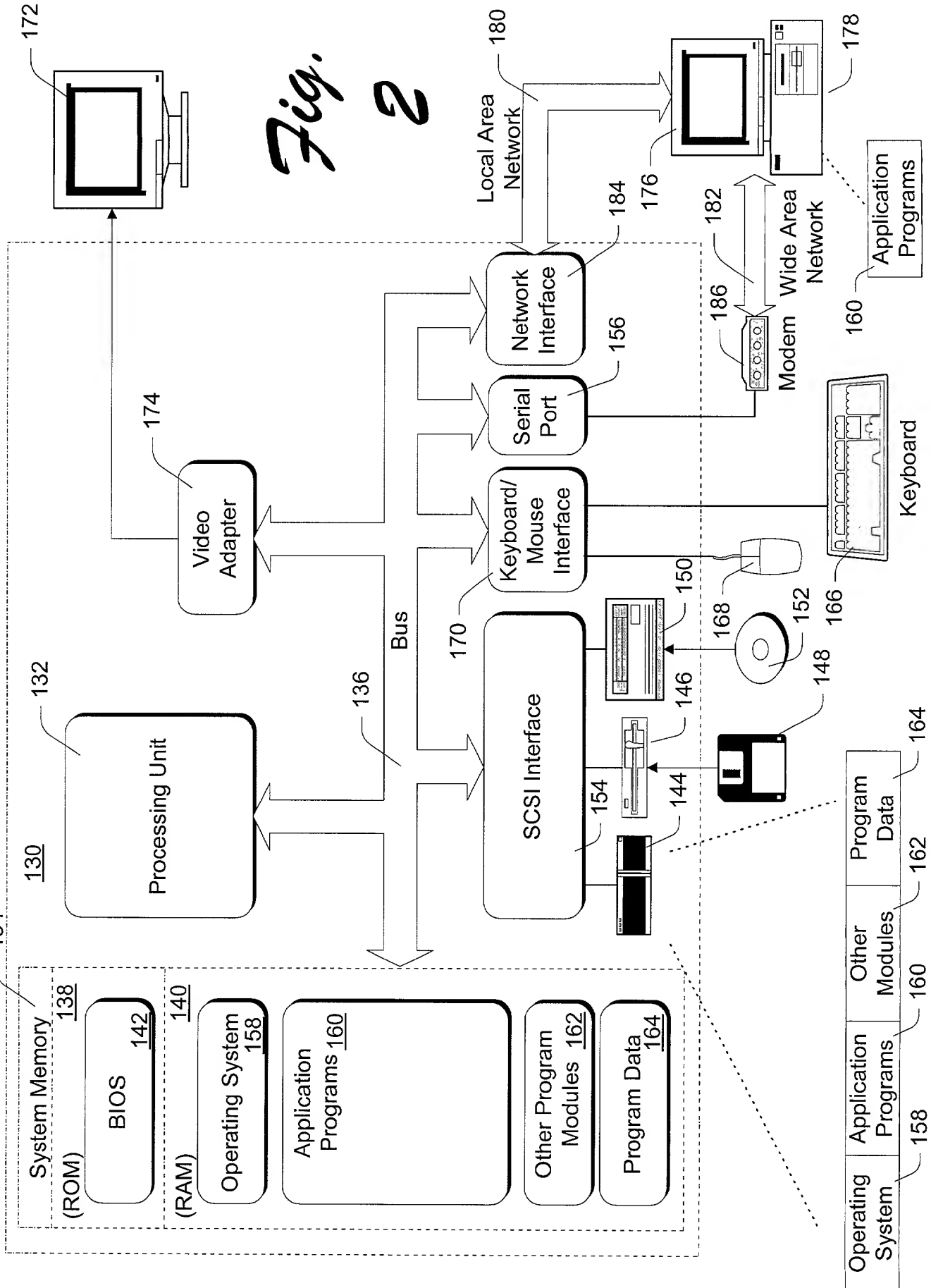
1 **31.** The collection of claim 26 configured for use by an ISAPI
2 extension.
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

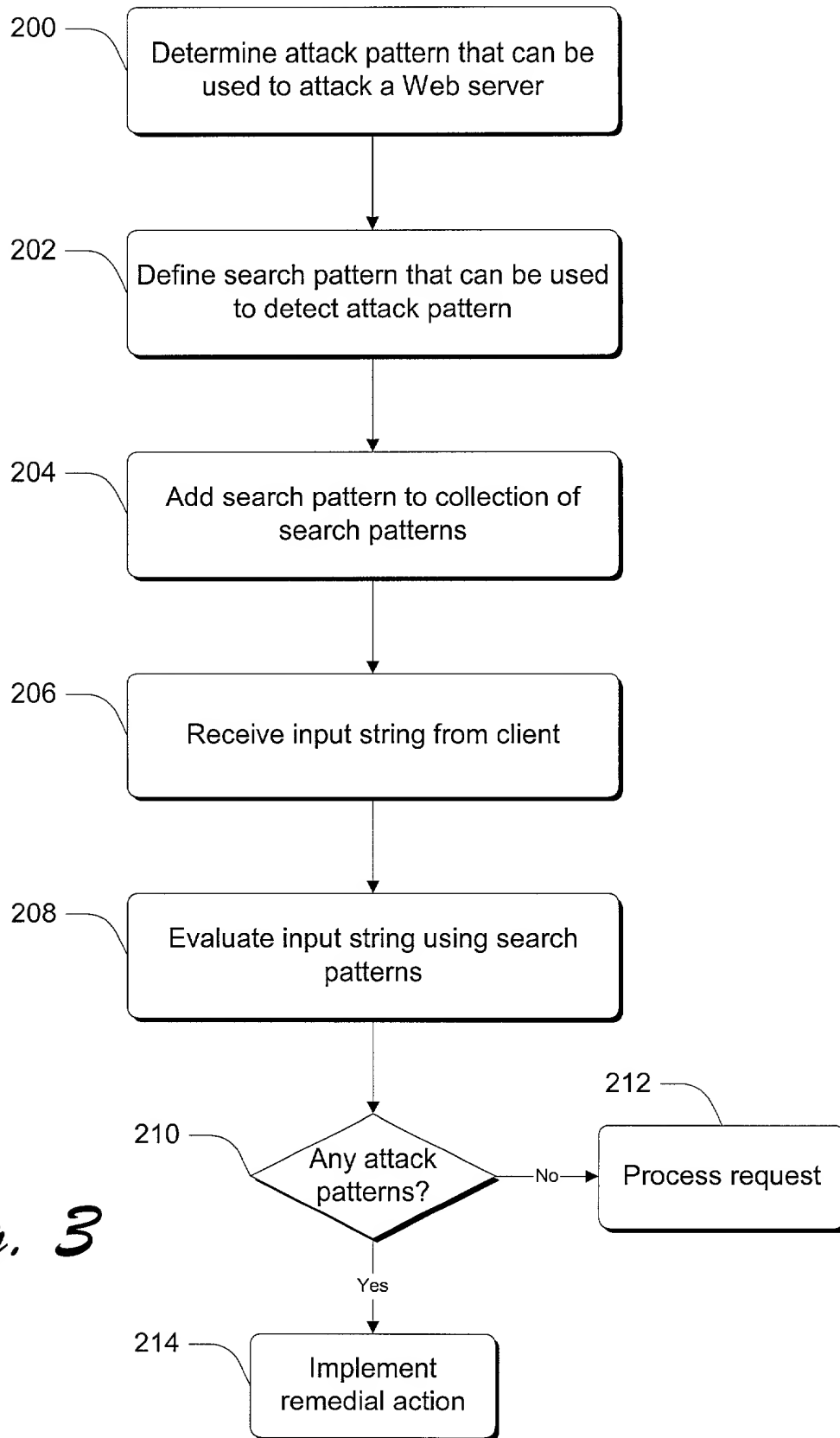
ABSTRACT

Methods and systems of screening input strings that are intended for use by a Web server are described. In the described embodiment, an attack pattern is determined that can be used to attack a Web server. A search pattern is defined that can be used to detect the attack pattern. The search pattern is defined in a flexible, extensible manner that permits variability among its constituent parts. An input string that is intended for use by a Web server is received and evaluated using the search pattern to ascertain whether the attack pattern is present. If an attack pattern is found that matches the search pattern, then a remedial action is implemented.

*Fig. 1*

134



*Fig. 3*

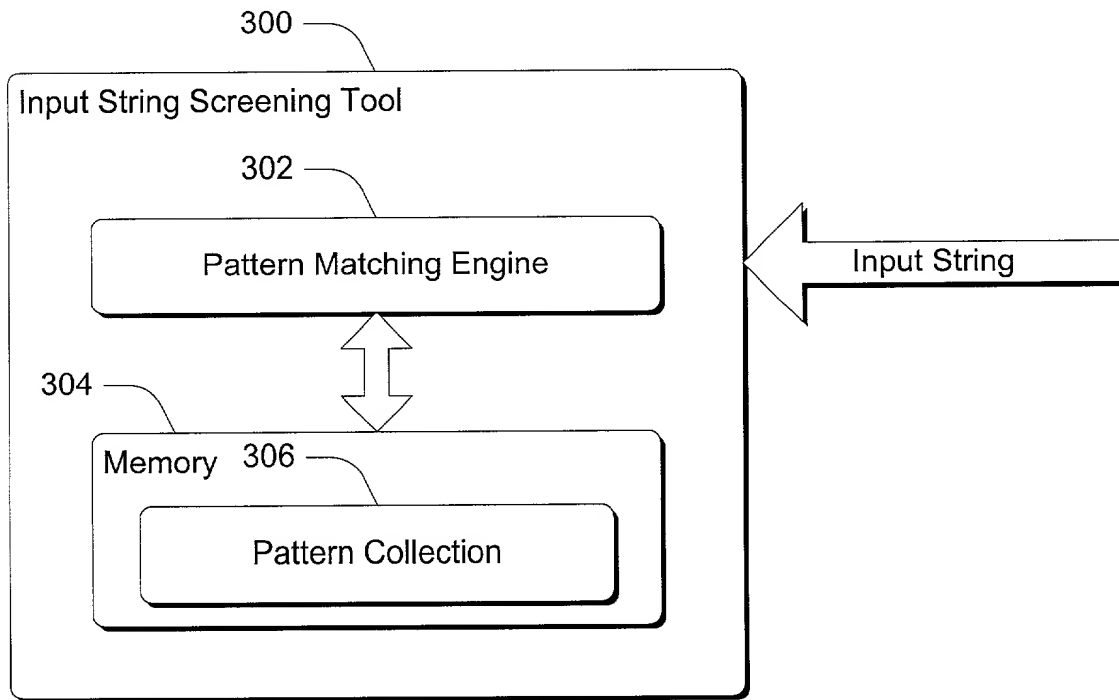


Fig. 4

1 **IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

2 Inventorship Howard et al.
 3 Applicant Microsoft Corporation
 4 Attorney's Docket No. MS1-379US
 Title: Methods and Systems for Screening Input Strings Intended for Use by Web
 Servers

5 **DECLARATION FOR PATENT APPLICATION**

6 As a below named inventor, I hereby declare that:

7 My residence, post office address and citizenship are as stated below next to
 8 my name.

9 I believe I am the original, first and sole inventor (if only one name is listed
 10 below) or an original, first and joint inventor (if plural names are listed below) of the
 11 subject matter which is claimed and for which a patent is sought on the invention
 12 entitled "Methods and Systems for Screening Input Strings Intended for Use by Web
 13 Servers," the specification of which is attached hereto.

14 I have reviewed and understand the content of the above-identified
 15 specification, including the claims.

16 I acknowledge the duty to disclose information which is material to the
 17 examination of this application in accordance with Title 37, Code of Federal
 18 Regulations, § 1.56(a).

19 PRIOR FOREIGN APPLICATIONS: no applications for foreign patents or
 20 inventor's certificates have been filed prior to the date of execution of this
 21 declaration.

22 **Power of Attorney**

23 I appoint the following attorneys to prosecute this application and transact all
 24 future business in the Patent and Trademark Office connected with this application:
 25 Lewis C. Lee, Reg. No. 34,656; Daniel L. Hayes, Reg. No. 34,618; Allan T.

1 Sponseller, Reg. 38,318; Steven R. Sponseller, Reg. No. 39,384; James R.
2 Banowsky, Reg. No. 37,773; Lance R. Sadler, Reg. No. 38,605; Michael A. Proksch,
3 Reg. No. 43,021; Thomas A. Jolly, Reg. No. 39,241; David A. Morasch, Reg. No.
4 42,905; Katie E. Sako, Reg. No. 32,628 and Daniel D. Crouse, Reg. No. 32,022.

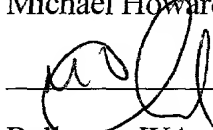
5 Send correspondence to: LEE & HAYES, PLLC, 421 W. Riverside Avenue,
6 Suite 500, Spokane, Washington, 99201. Direct telephone calls to: Lance R. Sadler
7 (509) 324-9256.

8
9 All statements made herein of my own knowledge are true and that all
10 statements made on information and belief are believed to be true; and further that
11 these statements were made with the knowledge that willful false statements and the
12 like so made are punishable by fine or imprisonment, or both, under Section 1001 of
13 Title 18 of the United States Code and that such willful false statement may
14 jeopardize the validity of the application or any patent issued therefrom.

15
16 * * * * *

17 Full name of inventor: Michael Howard

18 Inventor's Signature



Date: 11/8/99

19 Residence:

Bellevue, WA

20 Citizenship:

British

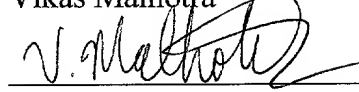
21 Post Office Address:

15798 SE 58th Place
Bellevue, WA 98006

Full name of inventor:

Vikas Malhotra

Inventor's Signature



Date: Nov. 08, 1999

Residence:

Brampton, Ontario

Citizenship:

Canadian

Post Office Address:

47 Newport Street
Brampton, Ontario, Canada L6S4N1